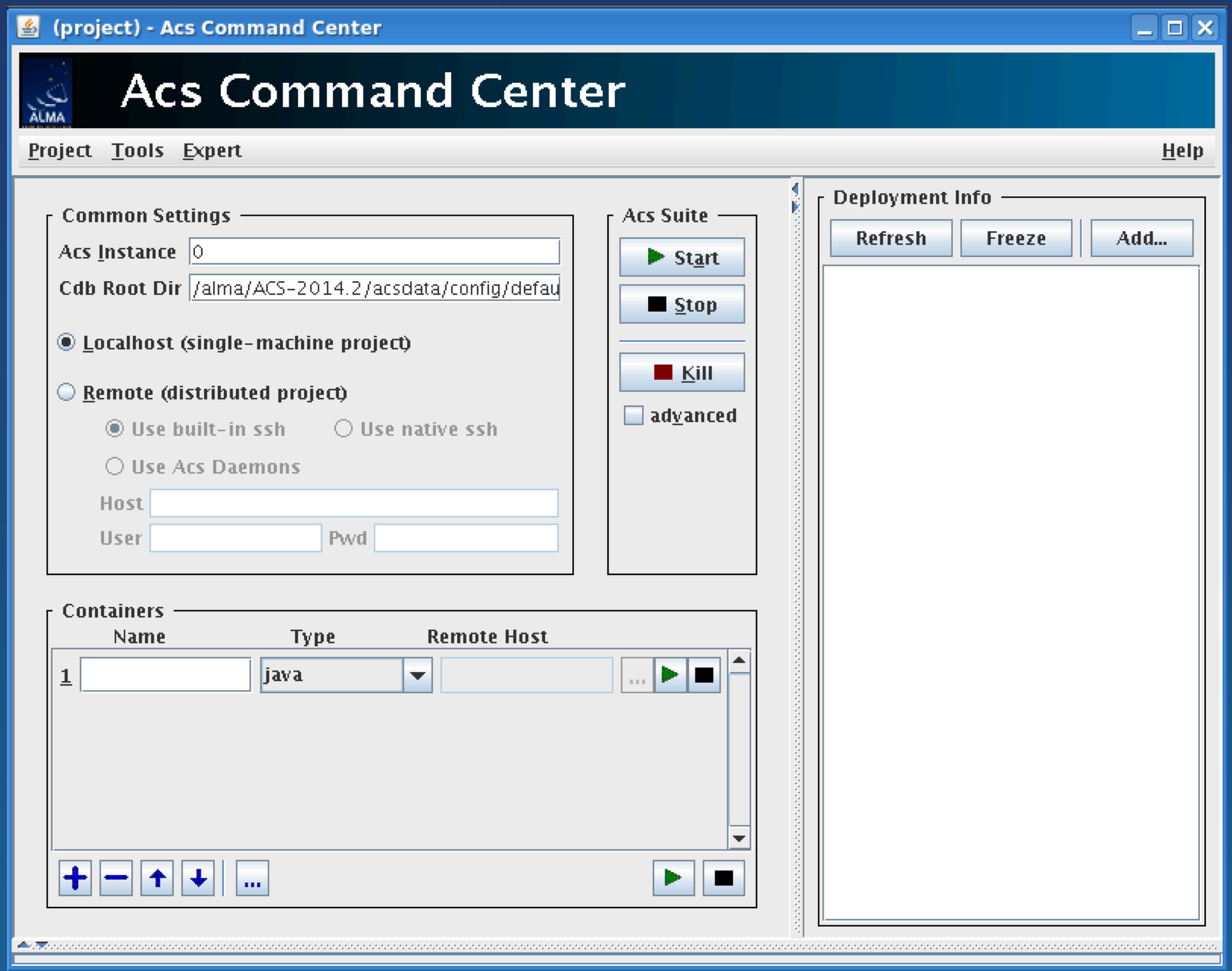
ALMA Common Software

Basic Track

A walk through ACS functionality

ACS Command Center: starting/stopping ACS



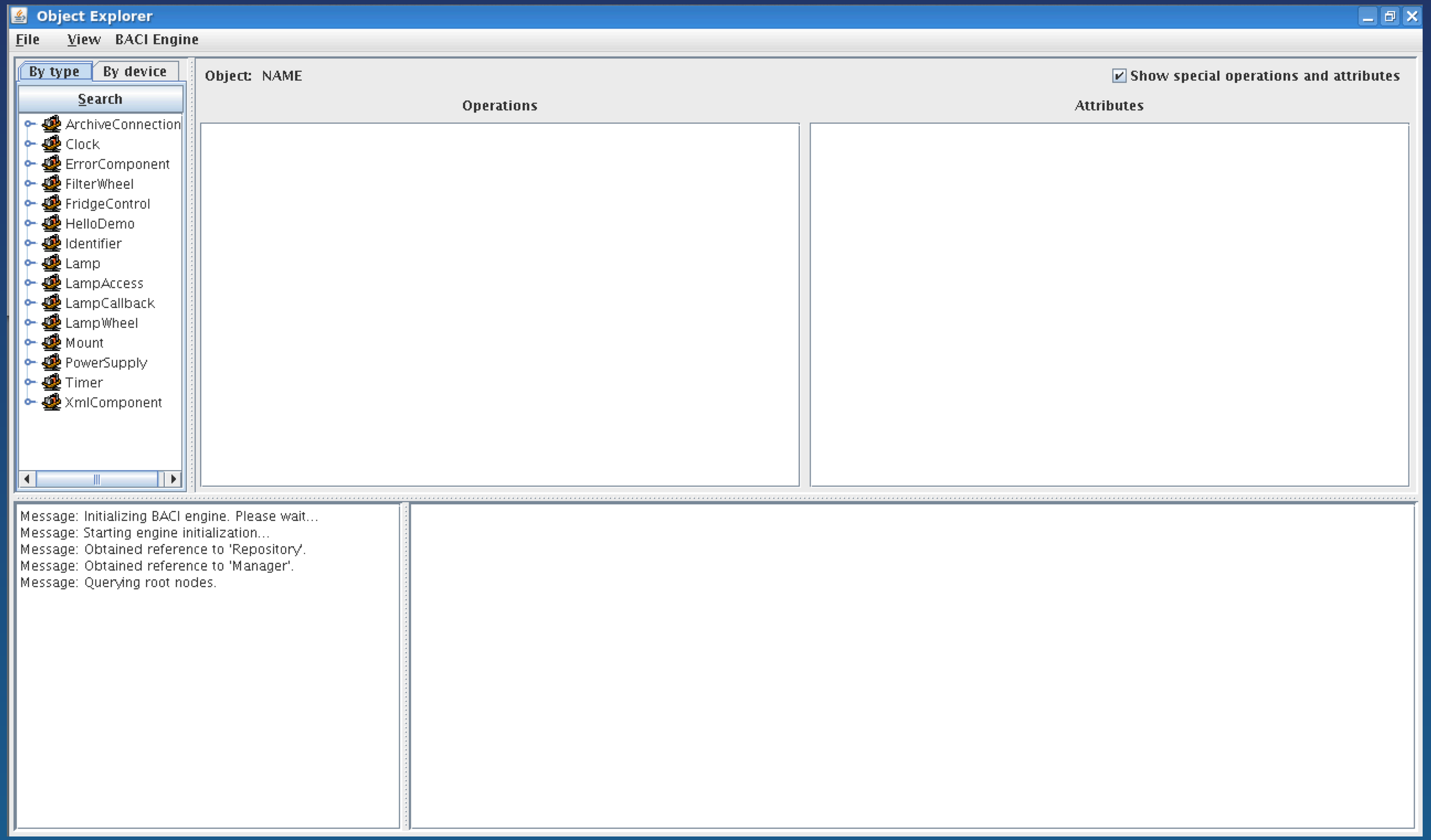
ACS Services

- Every application needs a set of **core** integration infrastructure services, like for instance:
  - messaging (request/response and publish/subscribe)
  - logging
  - error management
  - alarms
  - configuration data
- In ACS these services have been identified as essential for the application domain
- These have been implemented mostly on top of standard CORBA Services (DDS may replace CORBA in some/several of them)
- The ACS work consists in wrapping the implementation to simplify their usage by the application developers
- The following slides will present some of services provided by the ALMA Common Software

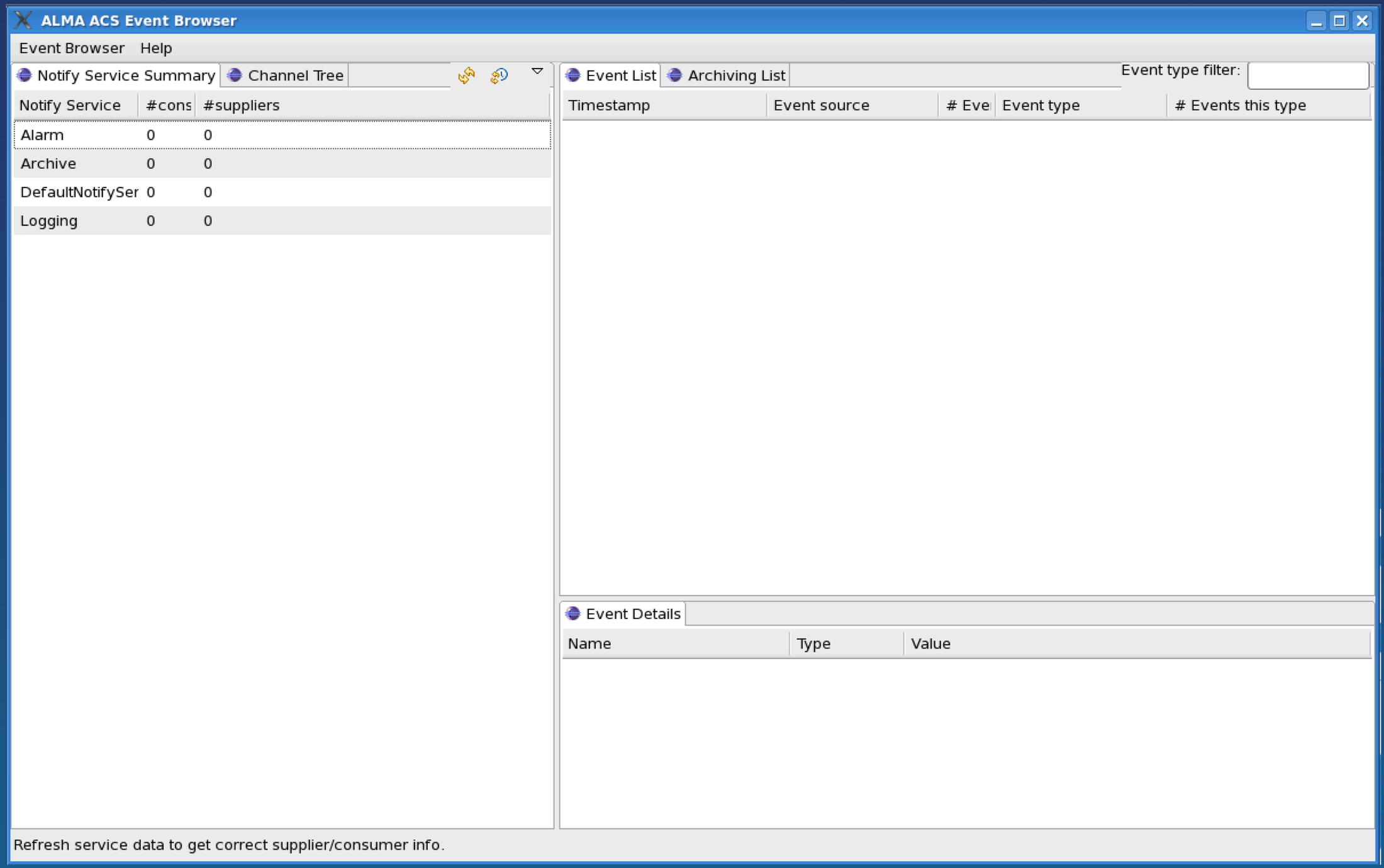
Messaging

- ✧ Enables the communication between system components
- ✧ Supports both request-response and publish-subscribe message patterns
  - ✧ anonymous publish/subscribe data transfer is seen more and more as a key need for the messaging system
- ✧ In ACS:
  - ✧ CORBA messaging provides request-response
  - ✧ CORBA Notify Service provides publish-subscribe
    - ✧ DDS can replace the Notify Service offering many advantages

Messaging tools: object explorer



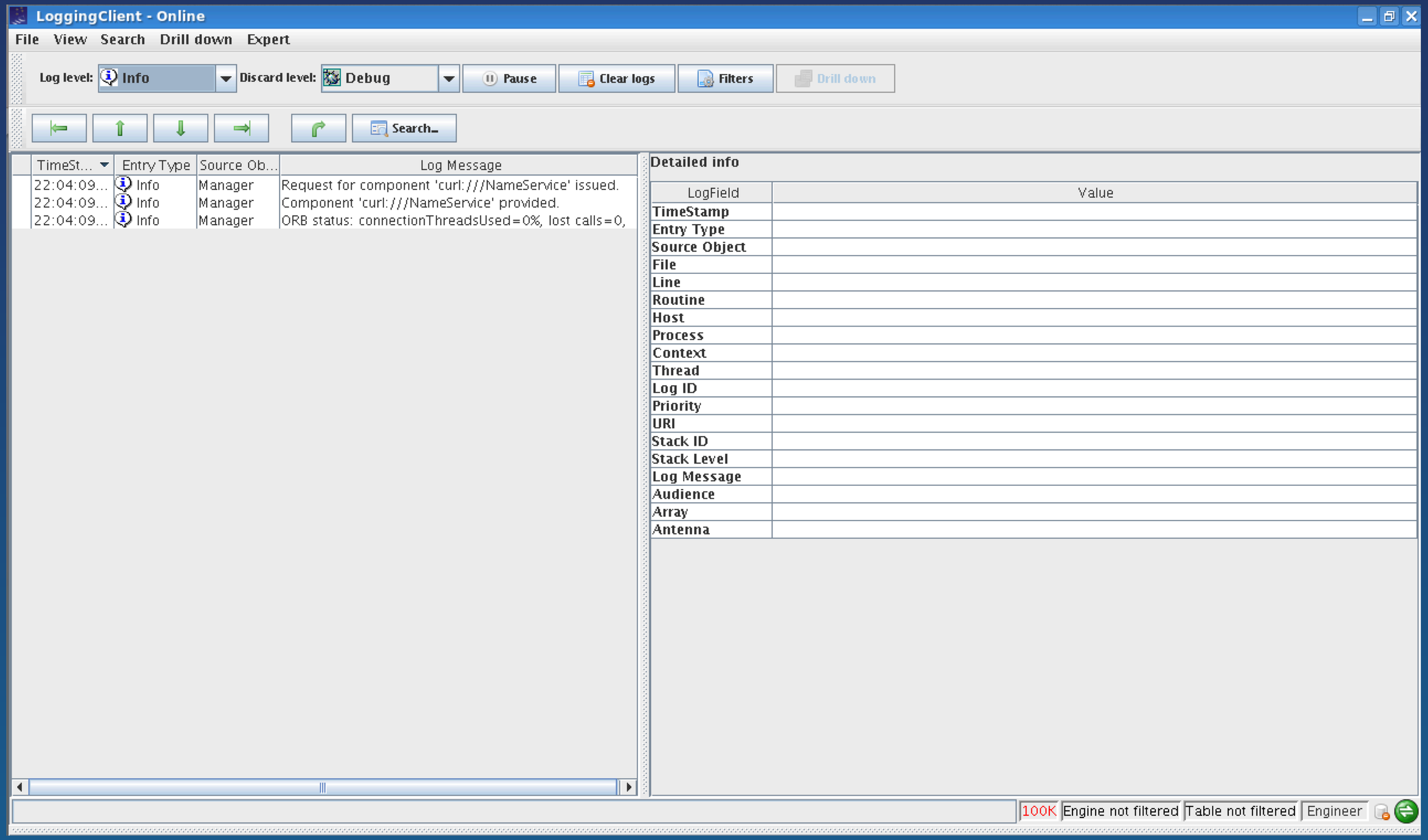
Messaging tools: event browser



Logging system

- ✧ Logging is fundamental for the operation of distributed systems, in order to understand and keep track of what happens between concurrent components
- ✧ Logging is used to publish any kind of status and diagnostic information for interested clients and for archival
- ✧ The current implementation is based on the Notification Service
- ✧ There is also a prototype implementation based on DDS

Logging tools: jlog logging client



Error System

- ✧ Provides a unified way of dealing with errors through the system
- ✧ CORBA supports “distributed” exceptions
- ✧ The ACS Error System provides additionally the following features:
  - ✧ Error format standardisation
  - ✧ Error handling design patterns
  - ✧ Error trace
  - ✧ Error logging
  - ✧ Synchronous and asynchronous error handling
  - ✧ Error browsing and definition tools

Alarm System

- ✧ Deals with **abnormal** situations
  - ✧ Fault states (FS)
  - ✧ Range from severe alarms to warning states
- ✧ Provides
  - ✧ FS collection, analysis and distribution, definition and archiving
  - ✧ FS reduction
  - ✧ Dedicated alarm consoles
- ✧ The ACS alarm system is a porting of the CERN LASER system

Configuration Database

- The ACS Configuration Database (CDB) addresses: defining, accessing and maintaining the configuration of a system
- For each component in the system, there might be a set of static (or quasi-static) configuration parameters that have to be configured in a persistent store and read when the component is started up or re-initialized.
- This includes the “deployment structure” of the system, i.e., which statically deployed Components are part of the system and their inter-relationships
- This information is used by the component/container infrastructure in runtime

Configuration Database Browser

Configuration Database Browser

File Edit Administration

Refrest CDB Tree

CURRENT LOCATION: /root/MACI/Containers/bilboContainer

- root
  - MACI
    - Containers
      - bilboContainer
        - Autoload
        - LoggingConfig
      - frodoContainer
      - aragornContainer
    - Managers
    - Channels
    - Components
  - alma
    - MOUNT3
    - TEST_PS_10
    - TEST_PS_1
    - TEST_PS_4
    - MOUNT5
    - LAMPWHEEL1
    - TEST_PS_14
    - MOUNT2
    - TEST_PS_18
    - TEST_PS_2
    - TEST_PS_9
    - PBUMP_B_01
    - TEST_PS_8
    - TEST_PS_17
    - PBUMP_B_02
    - TEST_PS_3

Save Changes to XML record | Reset Data

Table View | XML View

ATTRIBUTE NAME	ATTRIBUTE VALUE
DALtype	DAL
ImplLang	cpp
ManagerRetry	10
Recovery	true
ServerThreads	5
Timeout	20.0
UseIFR	1
xmlns	urn:schemas-cosylab-com:Container:1.0
xmlns:baci	urn:schemas-cosylab-com:BACI:1.0
xmlns:cdb	urn:schemas-cosylab-com:CDB:1.0
xmlns:log	urn:schemas-cosylab-com:LoggingConfig:1.0
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance

Sampling System

- ✧ Sampling of any Property
- ✧ High sustained frequency
- ✧ Optimized data transport
- ✧ Simultaneous sampling
- ✧ Plotting GUI

Component simulation

- Why simulation?
  - Distributed development
  - Features or entire subsystems not yet available
  - Test a subsystem in isolation
- Simulation of Components from IDL interface specification
- Dumb default or “intelligent” simulation

Questions?

Acknowledgements
ACS presentations were originally developed by the ALMA Common Software development team and has been used in many instances of training courses since 2004. Main contributors are (listed in alphabetical order): Jorge Avarias, Alessandro Caproni, Gianluca Chiozzi, Jorge Ibsen, Thomas Jürgens, Matias Mora, Joseph Schwarz, Heiko Sommer.

The Atacama Large Millimeter/submillimeter Array (ALMA), an international astronomy facility, is a partnership of Europe, North America and East Asia in cooperation with the Republic of Chile. ALMA is funded in Europe by the European Organization for Astronomical Research in the Southern Hemisphere (ESO), in North America by the U.S. National Science Foundation (NSF) in cooperation with the National Research Council of Canada (NRC) and the National Science Council of Taiwan (NSC) and in East Asia by the National Institutes of Natural Sciences (NINS) of Japan in cooperation with the Academia Sinica (AS) in Taiwan. ALMA construction and operations are led on behalf of Europe by ESO, on behalf of North America by the National Radio Astronomy Observatory (NRAO), which is managed by Associated Universities, Inc. (AUI) and on behalf of East Asia by the National Astronomical Observatory of Japan (NAOJ). The Joint ALMA Observatory (JAO) provides the unified leadership and management of the construction, commissioning and operation of ALMA.